IN THE CLAIMS:

The claims are as follows:

1. (Original) A method for performing alias refinement, the method comprising:

determining whether a load of an address exists for a variable in an intermediate representation of a source code;

determining, if the load of the address exists for the variable, whether each use of the address is for an indirect reference to the variable;

replacing, if a particular use of the address is for an indirect reference to the variable, the indirect reference in the intermediate representation with a direct reference to the variable; and

removing, if all uses of the address are for an indirect reference to the variable, the variable from an address taken alias set used with the intermediate representation.

2. (Original) The method of claim 1 wherein the address load determining, the use determining and replacing is repeated for each instruction in the intermediate representation.

3. (Original) The method of claim 1 further comprising:

creating a candidate list for the intermediate representation, where the candidate list contains the variable that requires the load of the address for the variable in the intermediate representation; and

removing, if one use of the address involves no indirect reference, the variable from the candidate list.

4. (Original) The method of claim 3 wherein the variable remaining on the candidate list is removed from the address taken alias set.

5. (Original) The method of claim 1 wherein the use of the address is represented with a pointer variable.

6. (Original) The method of claim 1 wherein the use of the address is represented in the intermediate representation with a load address command and a load of a value pointed by a pointer variable.

7. (Original) The method of claim 1 wherein the indirect reference in the intermediate representation comprises one of an indirect store of the variable to a memory and an indirect load of the variable from the memory.

8. (Original) The method of claim 1 wherein the indirect reference is a parameter in an inline procedure call.

9. (Original) The method of claim 1 wherein the use determining comprises:

propagating the uses of the address in the intermediate representation.

10. (Original) The method of claim 1 further comprising:

generating, after the replacing and the removing, the object code from the intermediate representation using the alias set; and

executing the object code.

11. (Previously Presented) A method for performing alias refinement, the method comprising:

determining whether a load of an address exists for a variable in an intermediate representation of a source code;

determining, if the load of the address exists for the variable, whether each use of the address is for an indirect reference to the variable and a particular use of the address is a parameter in a procedure;

replacing, if a particular use of the address is a parameter in a procedure call, in the intermediate reference the parameter with a direct reference to the variable; and

removing, if all uses of the address are for a parameter in a procedure call, the variable from an address taken alias set used with the intermediate representation.

12. (Original) An apparatus for performing alias refinement, the apparatus comprising:

a memory for storing a compiler program; and

a processor comprising a plurality of registers, where upon executing the compiler program, the processor is configured to:

determine whether a load of an address exists for a variable in an intermediate representation of a source code;

determine, if the load of the address exists for the variable, whether each use of the address is for an indirect reference to the variable;

replace, if a particular use of the address is for an indirect reference to the variable, the indirect reference in the intermediate representation with a direct reference to the variable; and

remove, if all uses of the address are for an indirect reference to the variable, the variable from an alias set used with the intermediate representation.

13. (Original) The apparatus of claim 12 wherein the processor determines the address load, determines the use, and replaces the indirect reference for each instruction in the intermediate representation.

14. (Original) The apparatus of claim 12 wherein the processor is further configured to:

create a candidate list for the intermediate representation, where the candidate list contains the variable that requires the load of the address for the variable in the intermediate representation; and

remove, if one use of the address involves no indirect reference, the variable from the candidate list.

15. (Original) The apparatus of claim 14 wherein the variable remaining on the candidate list is removed from the address taken alias set.

16. (Original) The apparatus of claim 12 wherein the processor is further configured to:

generate, after the replacing and the removing, the object code from the intermediate representation using the alias set; and

execute the object code.

17. (Original) A computer readable medium storing a software program that, when executed by a computer, causes the computer to perform a method comprising:

determining whether a load of an address exists for a variable in an intermediate representation of a source code;

determining, if the load of the address exists for the variable, whether each use of the address is for an indirect reference to the variable;

replacing, if a particular use of the address is for an indirect reference to the variable, the indirect reference in the intermediate representation with a direct reference to the variable; and

removing, if all uses of the address are for an indirect reference to the variable, the variable from an alias set used with the intermediate representation.

18. (Original) The computer readable medium of claim 17 wherein the address load determining, the use determining and replacing is repeated for each instruction in the intermediate representation.

19. (Original) The computer readable medium of claim 17 wherein the method further comprises:

creating a candidate list for the intermediate representation, where the candidate list contains the variable that requires the load of the address for the variable in the intermediate representation; and

removing, if one use of the address involves no indirect reference, the variable from the candidate list.

20. (Original) The computer readable medium of claim 19 wherein the variable remaining on the candidate list is removed from the address taken alias set.

21. (Original) The computer readable medium of claim 17 wherein the use of the address is represented with a pointer variable.

22. (Original) The computer readable medium of claim 17 wherein the use of the address is represented in the intermediate representation with a load address command and a load of a value pointed by a pointer variable.

23. (Original) The computer readable medium of claim 17 wherein the indirect reference in the intermediate representation comprises one of an indirect store of the variable to a memory and an indirect load of the variable from the memory.

24. (Original) The computer readable medium of claim 17 wherein the indirect reference is a parameter in a procedure call.

25. (Original) The computer readable medium of claim 17 wherein the use determining comprises:

propagating the uses of the address in the intermediate representation.

26. (Original) The computer readable medium of claim 17 wherein the method further comprises:

generating, after the replacing and the removing, the object code from the intermediate representation using the alias set; and

executing the object code.